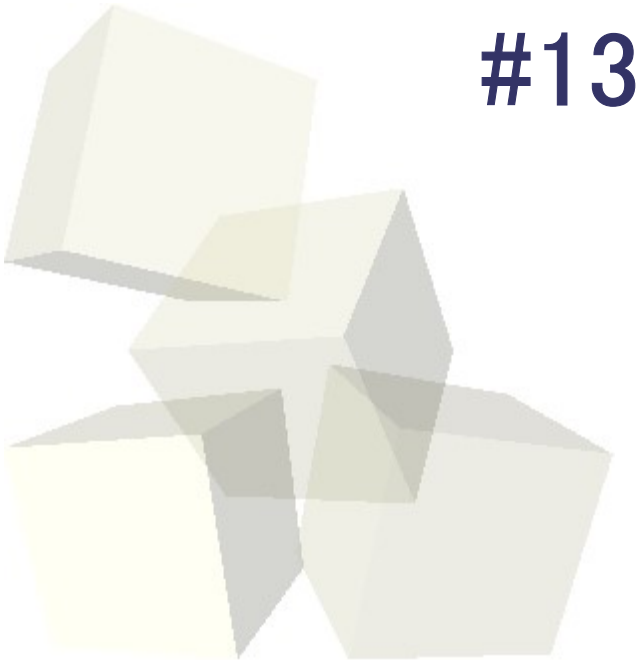
オペレーティングシステム

#13 ファイル：ファイル基礎



13.1
ファイルによる二次記憶の管理

記憶領域

	容量	揮発
レジスタ	小	揮発
主記憶	小	揮発
二次記憶	大	不揮発

ハードディスク,
磁気テープ(DATなど),
光ディスク(MO, CD, DVD),
フラッシュメモリ

アクセス方式

- 主記憶
  - アドレス
    - → 論理アドレス
    - → 物理アドレス

- 二次記憶
  - ファイルによる抽象化
    - → 管理すべき領域が膨大であるため
      主記憶におけるアドレスのような仕組みだけでは
      管理しにくい

ファイルの理想条件

- ファイル
  - 任意の時点で作成可能
  - 大きさを拡大・縮小可能
  - プロセス間で共有可能
  - 大きさに制限なし
    （二次記憶領域の大きさを超えない限り）

- ファイル名
  - 自由に設定可能
  - 同一ファイルを複数の名前で参照可能

13.2
二次記憶の種類とアクセス方法

デバイスとアクセス方式

テープ型デバイス	順次アクセス方式
ディスク型デバイス	直接アクセス方式

デバイスとアクセス方式

テープ型デバイス	順次アクセス方式
ディスク型デバイス	直接アクセス方式

テープ型デバイス

- 磁気テープ
  - オープンリール
  - CMT (Cartridge Magnetic Tape)
  - DLT (Digital Linear Tape)
  - LTO (Linear Tape-Open Ultrium)
  - QIC (Quarter Inch Cartridge)
  - Exabyte
  - DDS (Digital Data Storage)
  - etc...

ニュース

富士フイルムが新しい磁気テープ技術を開発、1巻で580TBが可能に

中田 敦　日経クロステック／日経コンピュータ

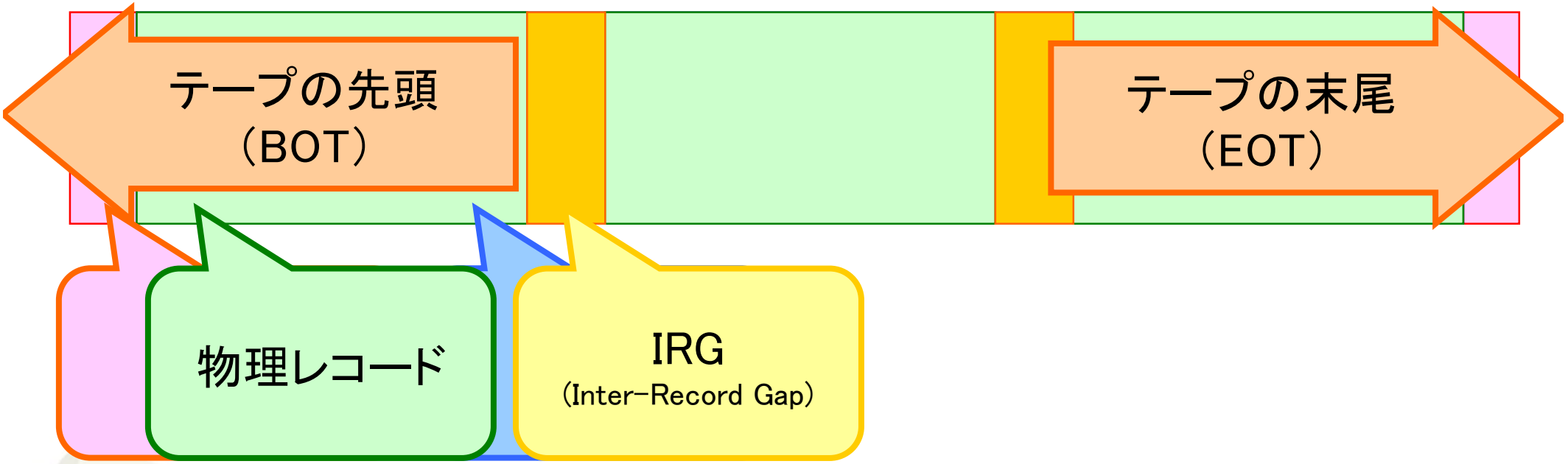
2020.12.16

富士フイルムは2020年12月16日、新たな磁性体として「ストロンチウムフェライト（SrFe）磁性体」を採用した磁気テープの実走行試験に成功したと発表した。SrFe磁性体によって記録密度が現行の「LTO-8」磁気テープに比べて約50倍に達し、1巻当たり580テラバイト（TB）の容量を備える磁気テープを開発できるようになるという。

https://xtech.nikkei.com/atcl/nxt/news/18/09343/

磁気テープの構造

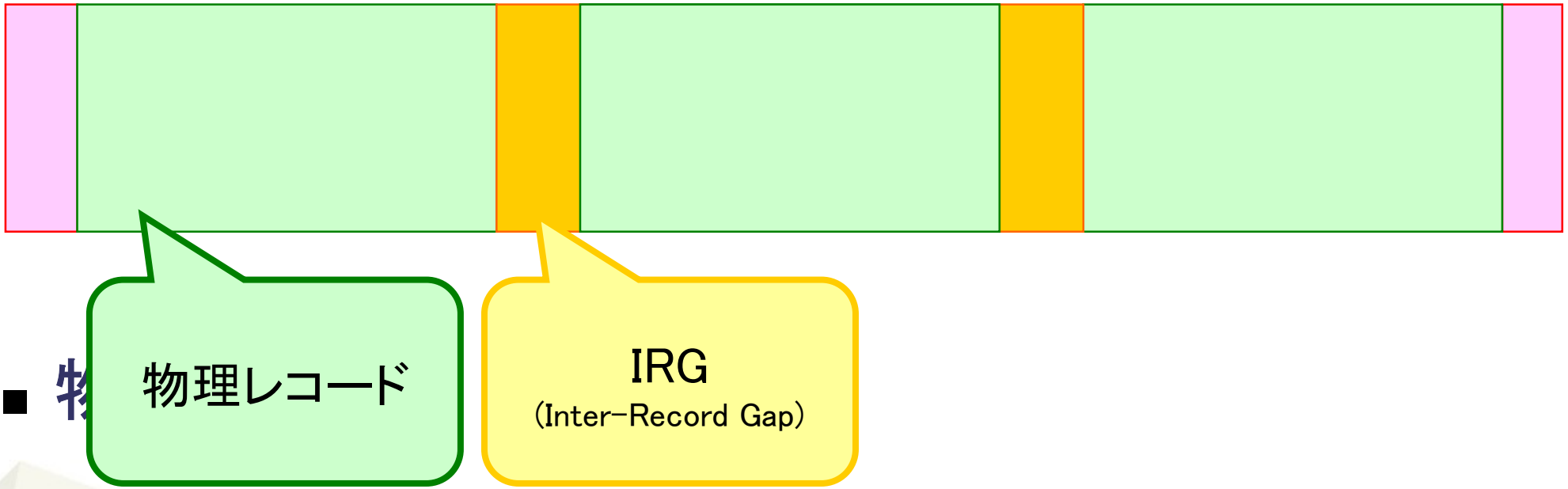
テープ上の物理的なデータ配置



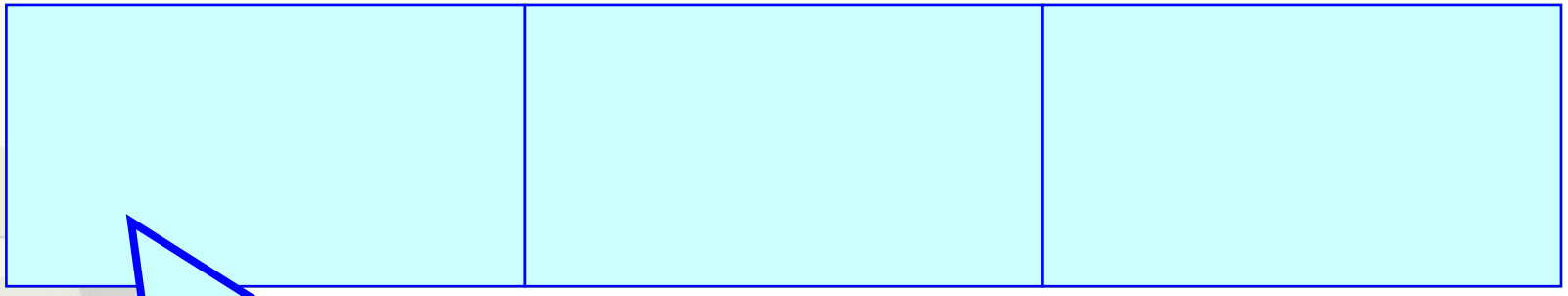
- 物理レコード
  - → 1処理でテープから読み出す単位
- IRG (Inter-Record Gap, aka IBG)
  - → レコード間に置く空白領域

磁気テープの構造

- テープ上の物理的なデータ配置

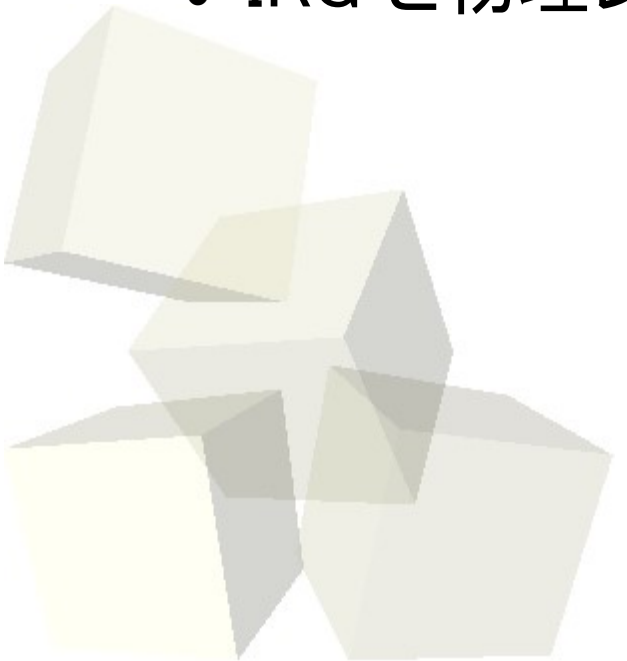


- 物



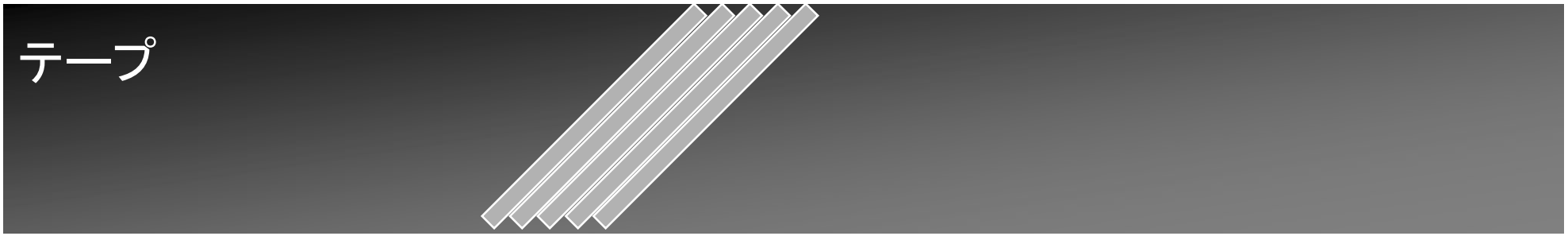
IRG

- IRG
  - 物理レコード間の空白領域
  - IBG (Inter-Block Gap) と呼ぶことも
- 物理レコード長が記憶容量に影響
  - IRG と物理レコードの比率が変化するため



DATなど（ヘリカルスキャン）

- テープ上の物理的なデータ配置



- 論理的なデータの並び

  - 短いテープ長，遅い回転数で多くのデータを格納可能

余談：ヘリカルスキャン方式は、東芝の澤崎憲一博士により開発された。
https://toshiba-mirai-kagakukan.jp/learn/history/ichigoki/1959vtr/index_j.htm

順次アクセス方式

- 磁気テープ
  - データの並びは一次元
  - 連続的にデータを読み出す（順次アクセス）に向く
  - 特定のファイルを探すような場合，テープの最初からずっと見ていかなければならない

- テープにおいて可能な処理
  - テープヘッドのある位置から物理レコードを読み出す
  - テープの先頭（BOT）まで巻き戻す
  - ファイルの終わり（EOF，次のファイルの先頭）までテープヘッドを進める

デバイスとアクセス方式

テープ型デバイス	順次アクセス方式
ディスク型デバイス	直接アクセス方式

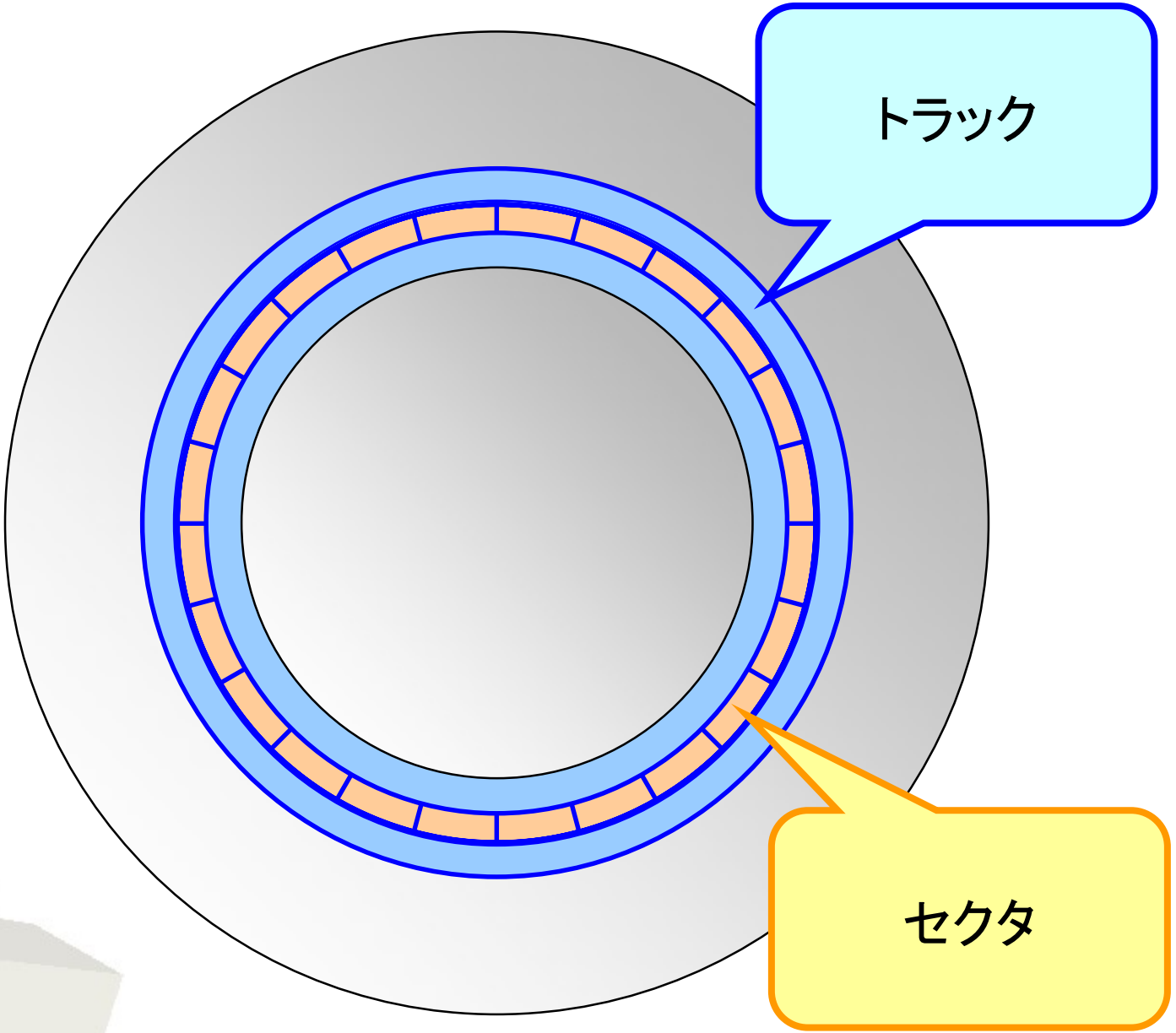
ディスク型デバイス

- 磁気ディスク
  - Floppy Disk
  - Hard Disk

- 光磁気ディスク
  - MO

- 光ディスク
  - CD
  - DVD

ディスク



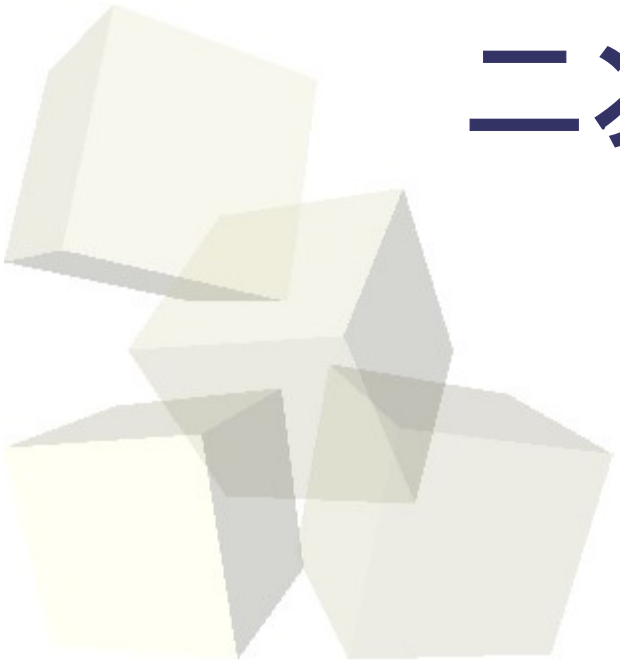
ハードディスク上の位置指定

- ボリューム
  - ハードディスクドライブの指定
- シリンダ番号
  - 複数ディスク上の同一半径トラックの集合
- ヘッド番号
  - ヘッドの番号（ディスク枚数 x 2）
- セクタ番号

シーク速度

- ハードディスクへのアクセス
  - 任意の場所を直接指定可能(直接アクセス)
- アクセスの手順
  - シリンダ(ディスク)を回転
  - ヘッドを回転し目的セクタへ
- アクセス速度(シーク速度)
  - 非常に遅い:数msから数十ms
    - → 参考:CPUの1命令の動作速度は 0.5から1ns

13.3
プログラム側から見た
二次記憶アクセス方式



アクセス方式の分類

- ファイル構造を意識する場合
  - 順編成
  - 直接編成
  - 区分編成
  - 索引順編成
- ファイル構造を意識しない場合
  - ストリーム型入出力

アクセス方式の分類

- ファイル構造を意識する場合
  - 順編成
  - 直接編成
  - 区分編成
  - 索引順編成
- ファイル構造を意識しない場合
  - ストリーム型入出力

順編成と直接編成

- 順編成
  - 順次アクセス
  - ヘッドの位置を表すポインタ変数を用いる
- 直接編成
  - 直接アクセス
  - セクタ単位での読み書き
- 区分編成
  - ファイルを複数の仮想的な区分（メンバ）に分割
  - 区分単位で読み書き

アクセス方式の分類

- ファイル構造を意識する場合
  - 順編成
  - 直接編成
  - 区分編成
  - 索引順編成
- ファイル構造を意識しない場合
  - ストリーム型入出力

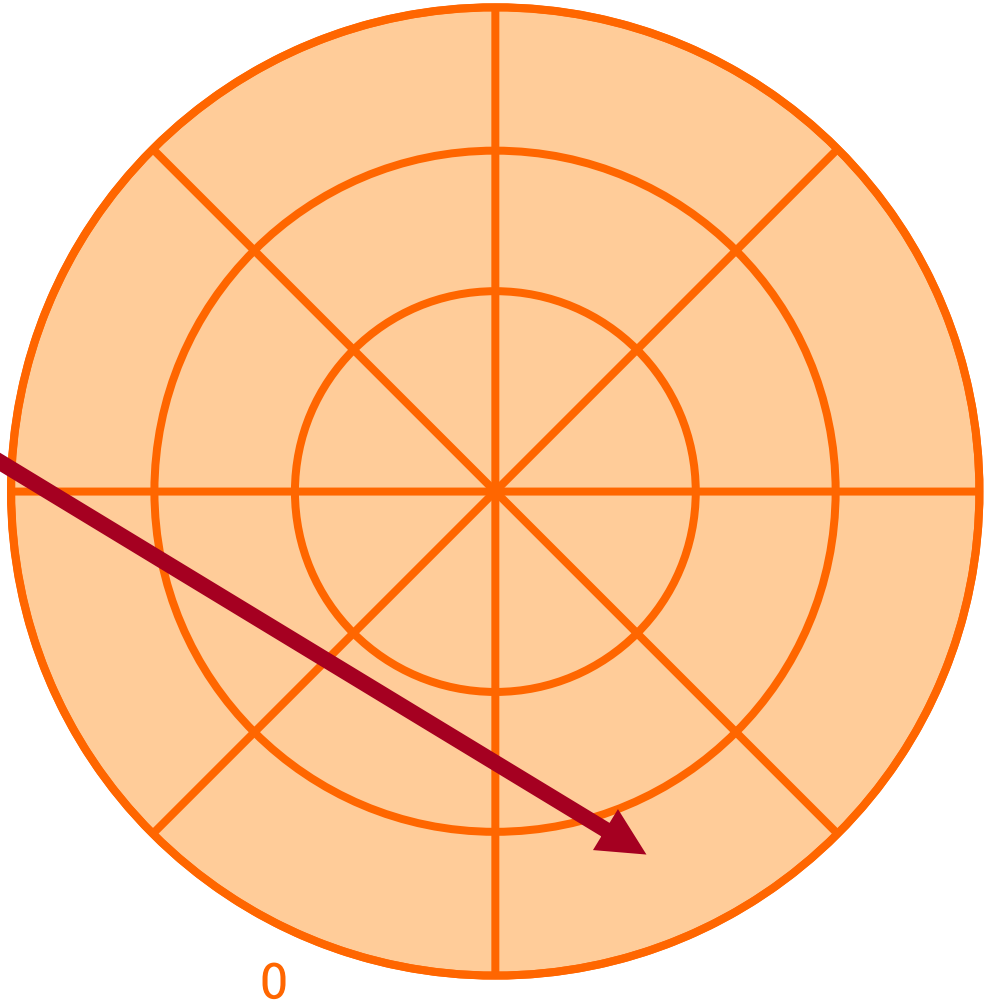
索引順編成

- ファイル名でなく検索語でファイル位置を指定

検索
「N」

ファイルの内容を代表するような値

ID	ポインタ（トラック，セクタ）
N3	(1, 7)
P5	(2, 1)
A6	(2, 2)
Q9	(1, 1)
:	:



アクセス方式の分類

- ファイル構造を意識する場合
  - 順編成
  - 直接編成
  - 区分編成
  - 索引順編成

- ファイル構造を意識しない場合
  - ストリーム型入出力

ストリーム型入出力

- 「ファイル構造を意識しない」とは
  - ファイルの読み書き時に
    レコードやセクタを意識しない
  - ファイルは区切りのないバイト列として読み書き

ストリーム型入出力

- ストリーム型入出力
  - Windows, UNIX などで採用
  - ファイルの先頭 or ファイルポインタの位置に
    読み書き操作可能
  - バイト単位での読み書きが可能

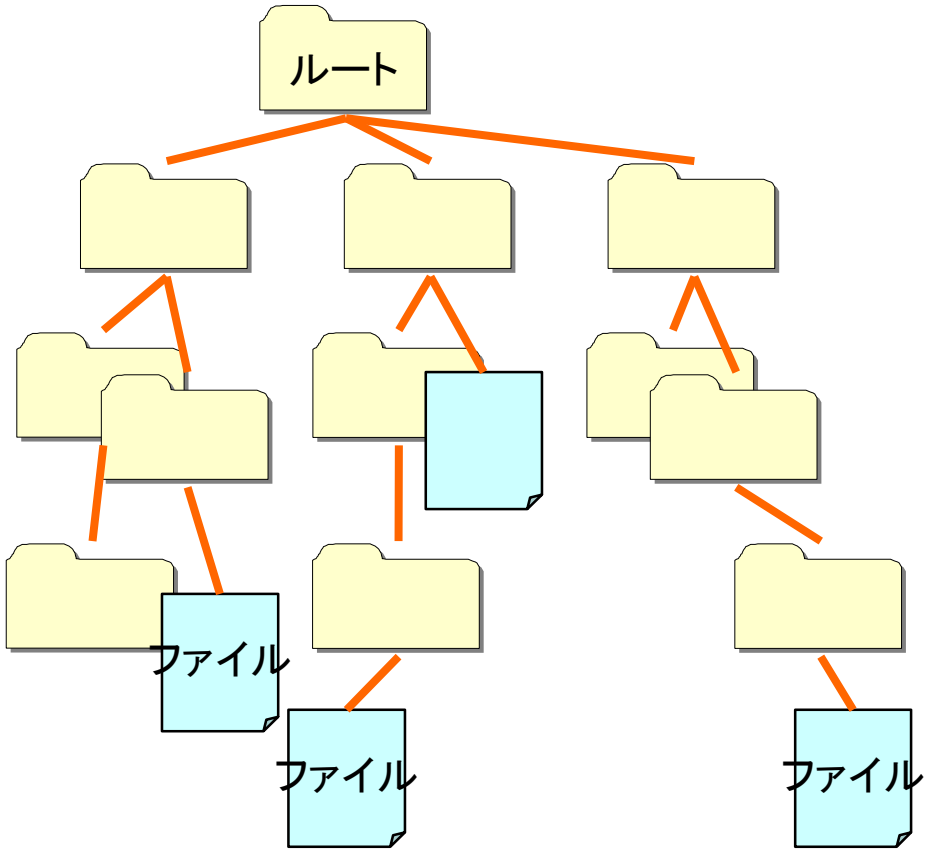
13.4
階層化ディレクトリシステム

階層化ディレクトリシステム

- ファイル名重複の問題
  - 特に複数ユーザで利用する場合など，
    同一ファイル名を使用したい場合
  - ファイルを構造的に管理したい
  - 「ディレクトリ」と呼ぶ「容れ物」で管理
- 参考：現実世界における保管場所
  - 洋服：「自分の家の」「自分の部屋の」「タンスの」「2番目の引き出し」
  - 保管場所はある「容れ物」であり，その容れ物はさらに大きな「容れ物」内に位置している
  - 「自分の家」も，
    「自分の町」などの更に大きな容れ物の中

階層化ディレクトリシステム

- 「入れ子構造」
  - 便宜的に木構造で描くことが多い
  - 最大の「容れ物」は木構造における木の根に位置するため「ルートディレクトリ」と呼ばれる



UNIXにおけるパス（path）

- ファイルの格納場所を指定する書式
  - 絶対パス
    - → ルートディレクトリからの経路
  - 相対パス
    - → 他のファイルやディレクトリ（一般にはカレントディレクトリ）からの経路
- 特殊記法
  - . ：カレントディレクトリ
  - .. ：親ディレクトリ
  - ~ ：ホームディレクトリ

絶対パスと相対パス

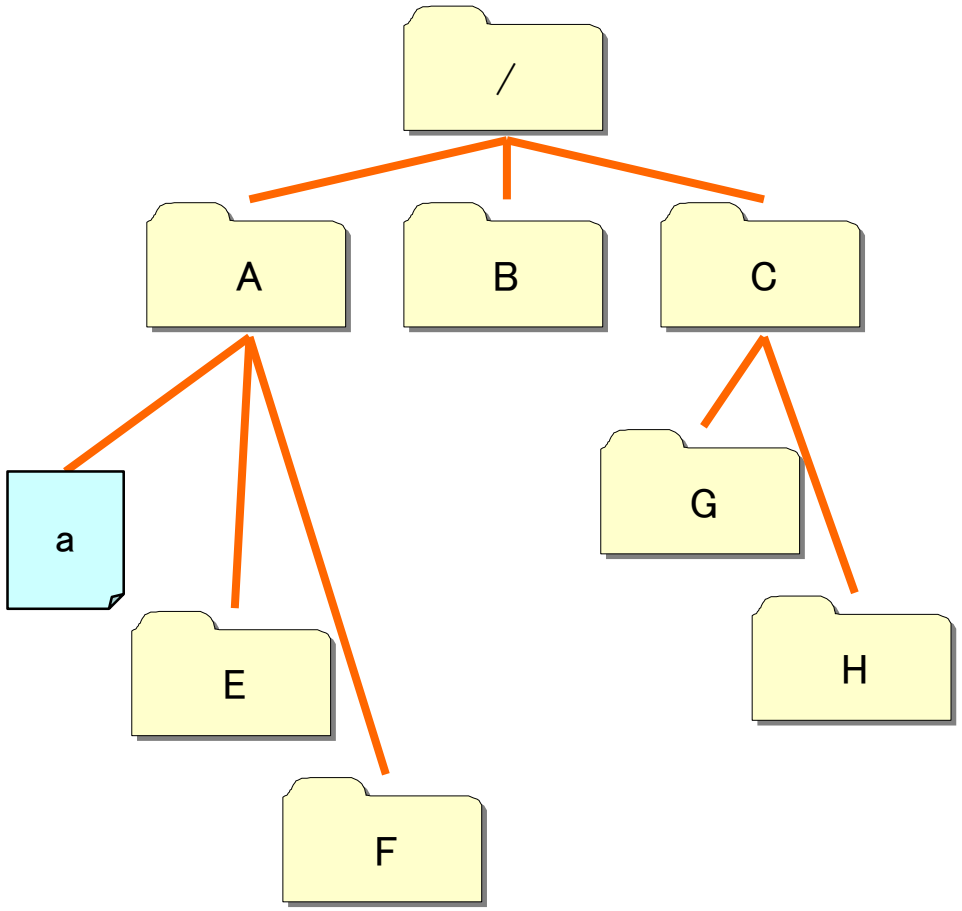
- Aがカレントディレクトリの場合の
  ファイルaのパス表記

絶対パス

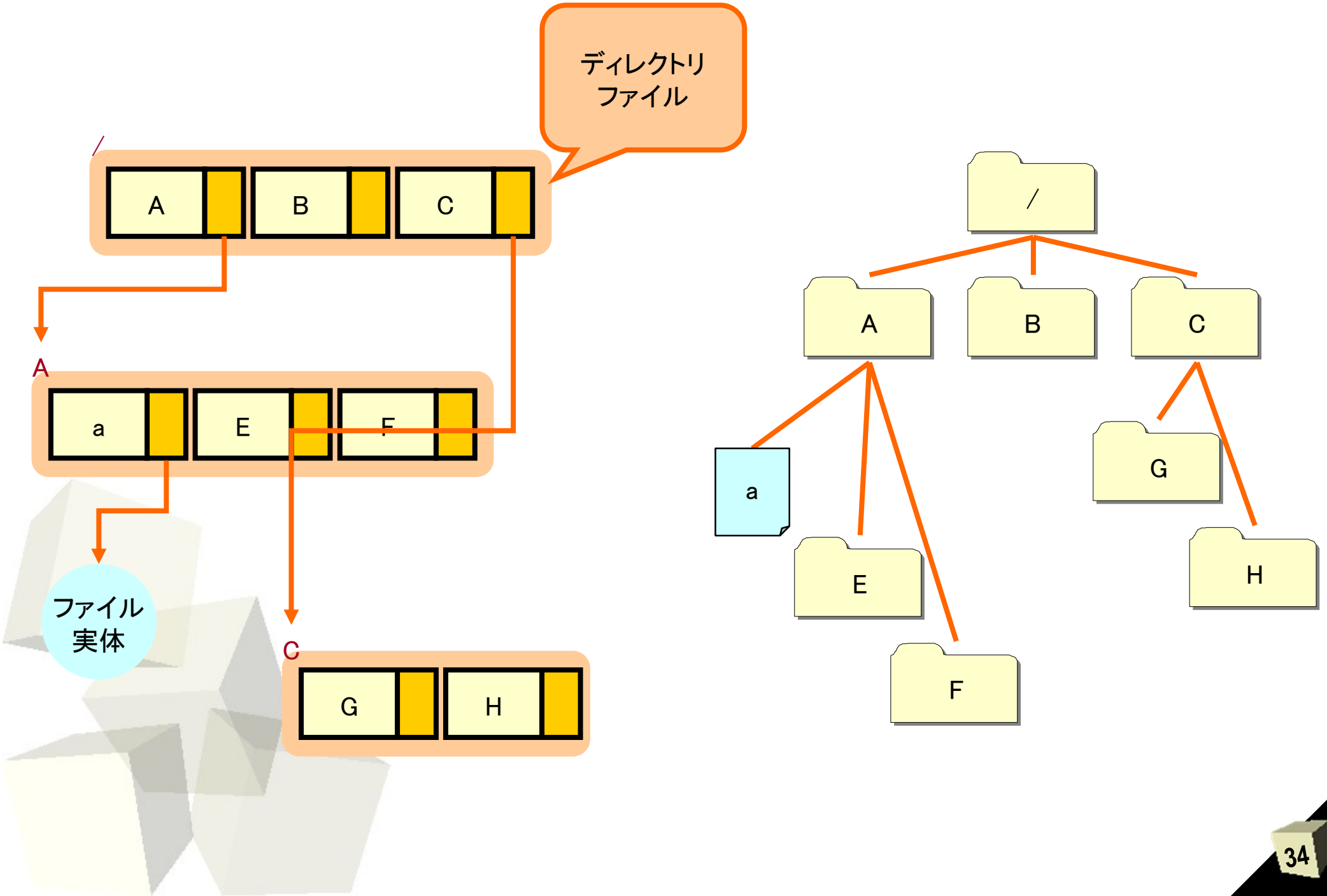
- /A/a

相対パス

- a
- ./a
- ../A/a



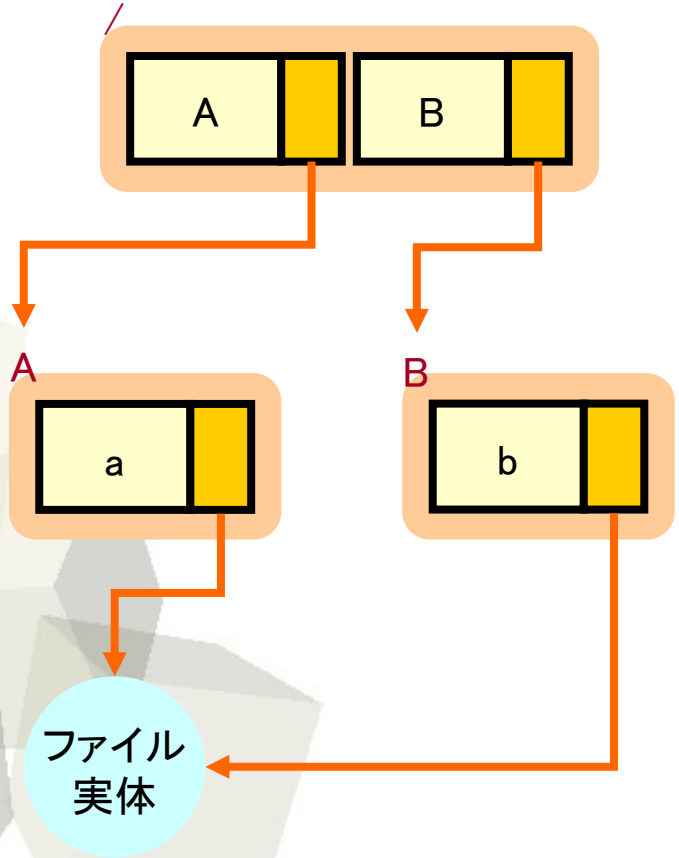
絶対パスと相対パス

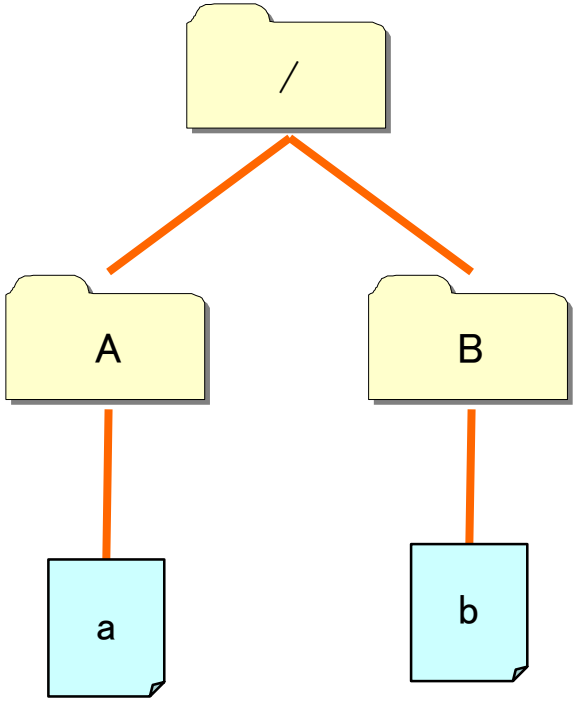


リンク

- 同一ファイルを複数名で参照可能にするしくみ

- ハードリンク

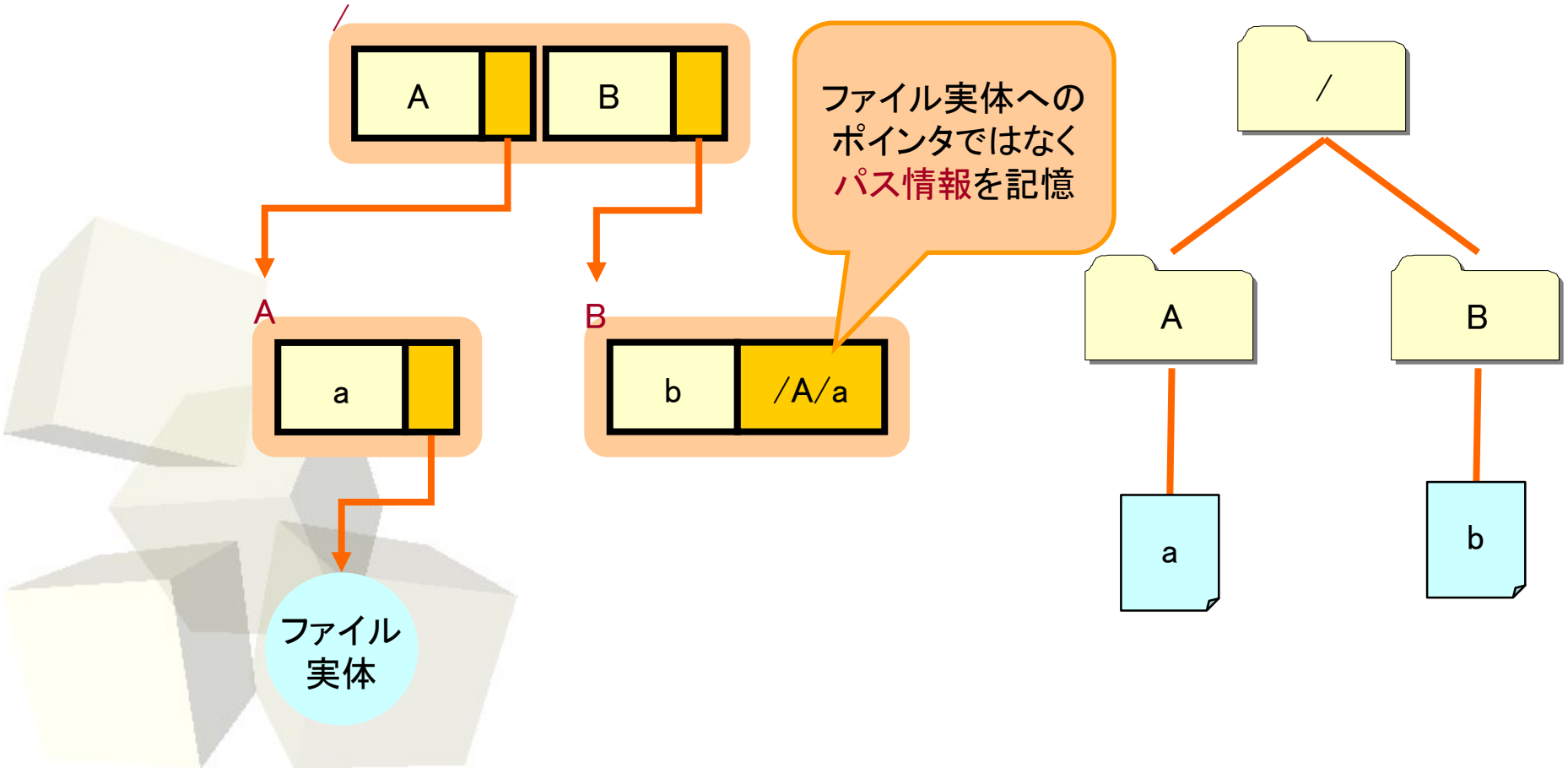




リンク

- 同一ファイルを複数名で参照可能にするしくみ

- シンボリックリンク



リンク

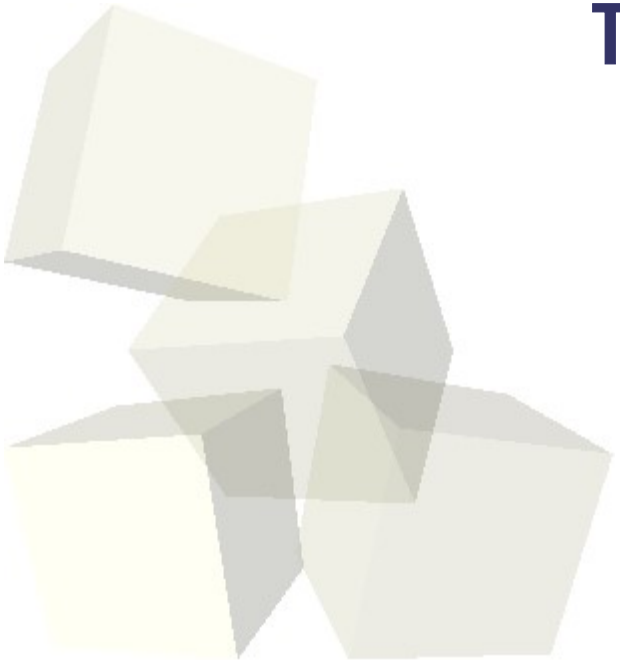
ハードリンク

- ファイル実体を複数のディレクトリファイルが指す
- 全てのファイル名に対し削除が行われると
  ファイルは削除される

シンボリックリンク

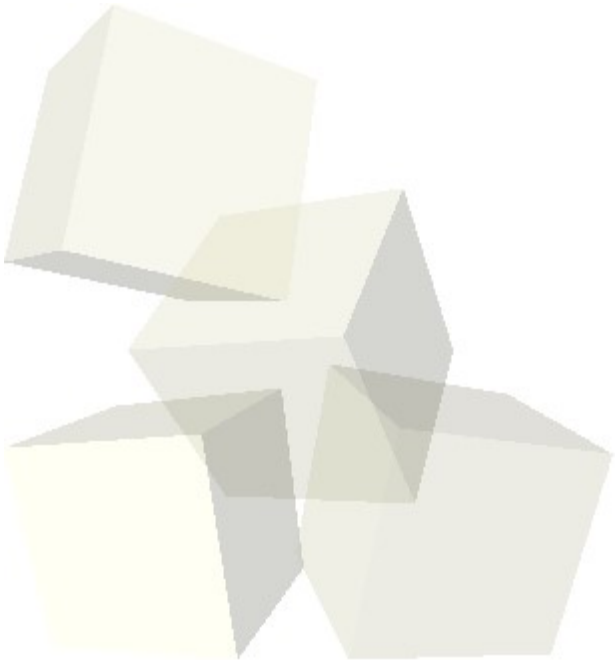
- ファイル実体へのポインタではなく,
  ファイル名のパス情報により参照
- ファイルが削除された場合,
  シンボリックリンクだけが残ってしまう
  （参照先の実体が存在しないリンクに）

13.5
領域割り当て方式



領域割り当て

- ファイルに対する領域の割り当て
  - ディスク型の場合
  - 「プロセスに対する主記憶領域の割当て」と同様
    「ファイルに対する二次記憶領域の割当て」が必要



領域割り当て方式

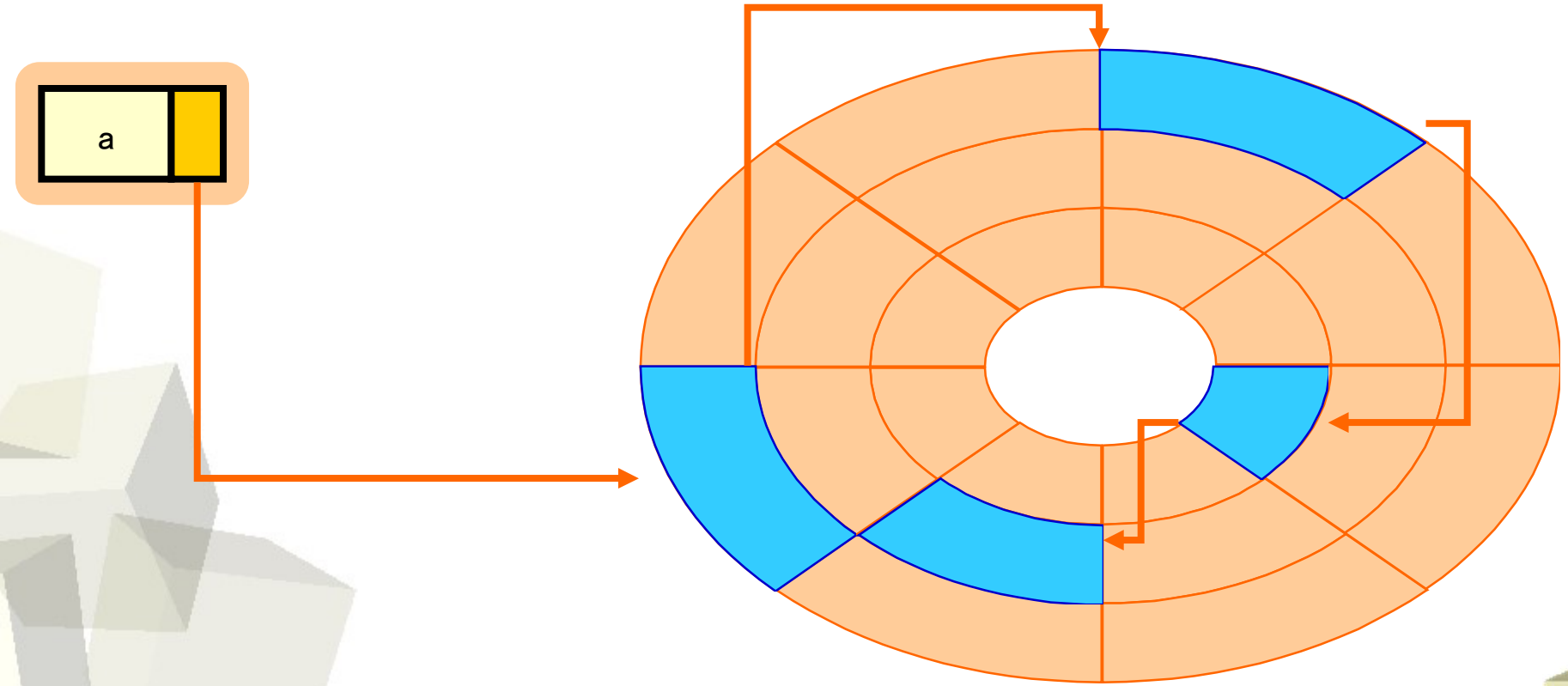
- 固定長割り当て方式
  - → プロセスが必要とする量の二次記憶領域を
    1セクタまたは1クラスタ（複数セクタをまとめた単位）
    単位で割り当てる
  - 複数の単位領域の集合を一領域として扱う仕組が必要
    - → リスト方式
    - → インデクス方式
- 連続領域割り当て方式
  - → プロセス実行前に，
    必要となる大きさの連続したセクタを割り当てる

領域割り当て方式

- 固定長割り当て方式
  - → プロセスが必要とする量の二次記憶領域を
    1セクタまたは1クラスタ（複数セクタをまとめた単位）
    単位で割り当てる
  - 複数の単位領域の集合を一領域として扱う仕組が必要
    - → リスト方式
    - → インデクス方式
- 連続領域割り当て方式
  - → プロセス実行前に，
    必要となる大きさの連続したセクタを割り当てる

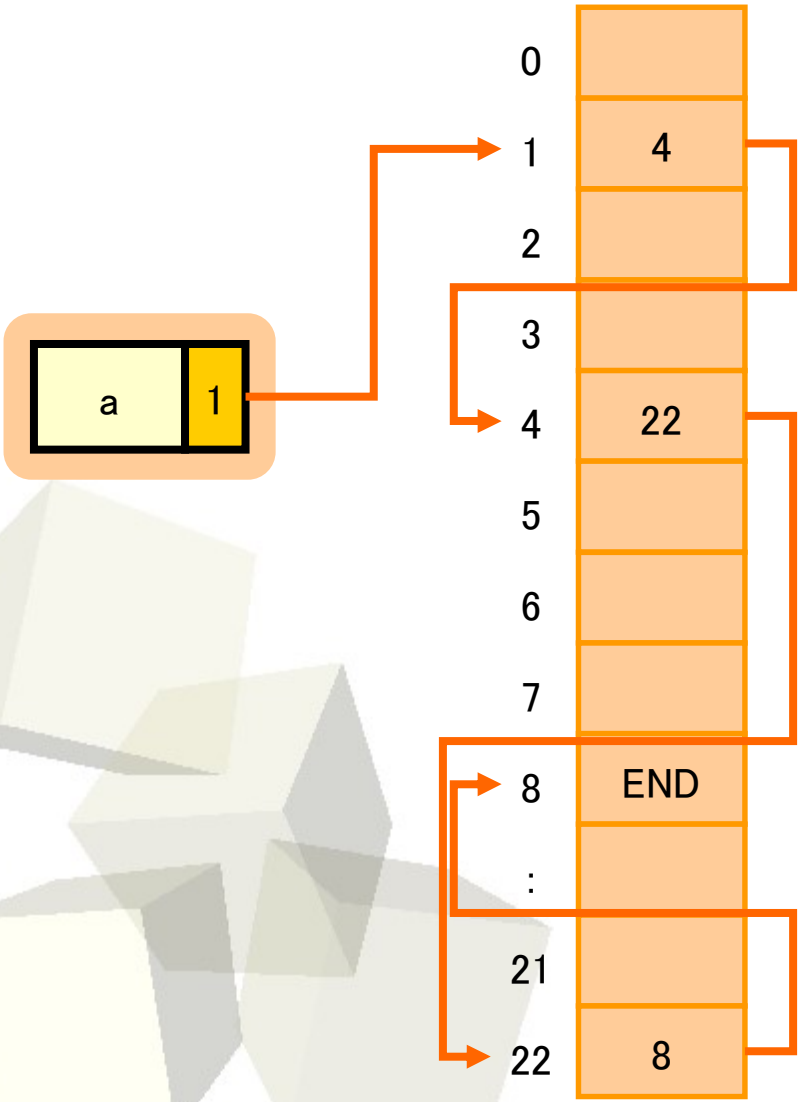
リスト方式

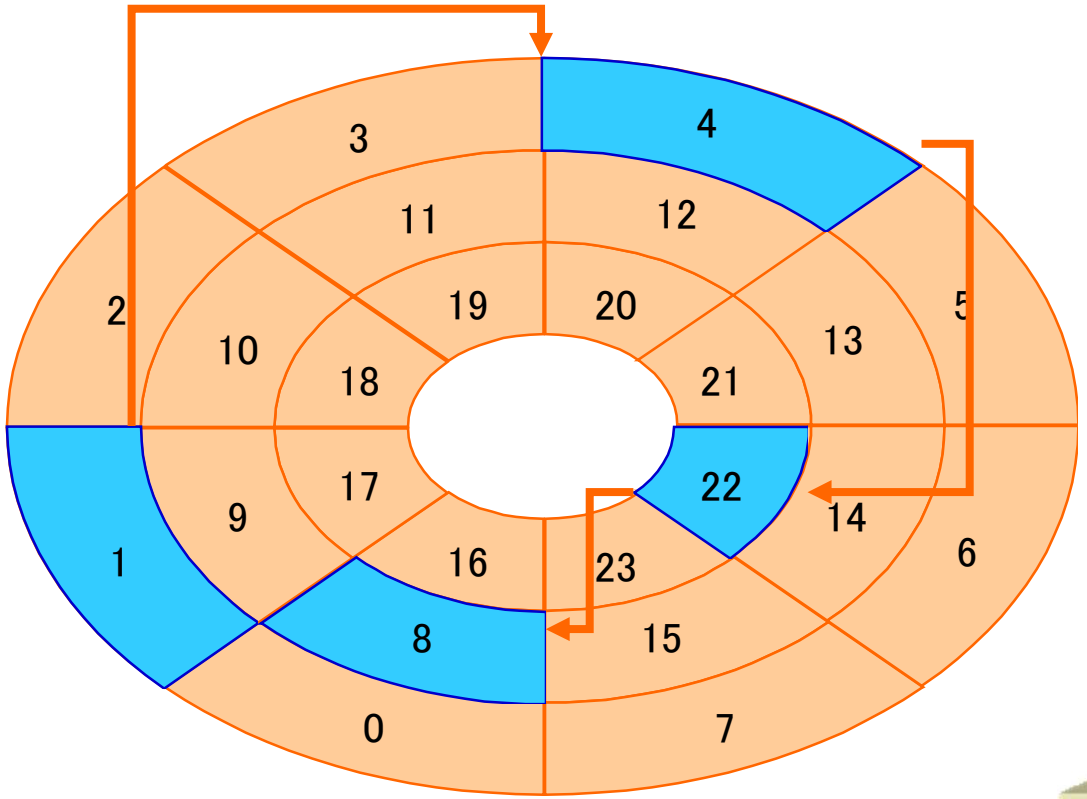
- ファイル構成セクタをリストで管理
  - ディレクトリファイルは先頭構成セクタへのポインタを保持
  - セクタの最後に次のセクタへのポインタを格納



インデクス方式

- セクタの連結をテーブルで管理





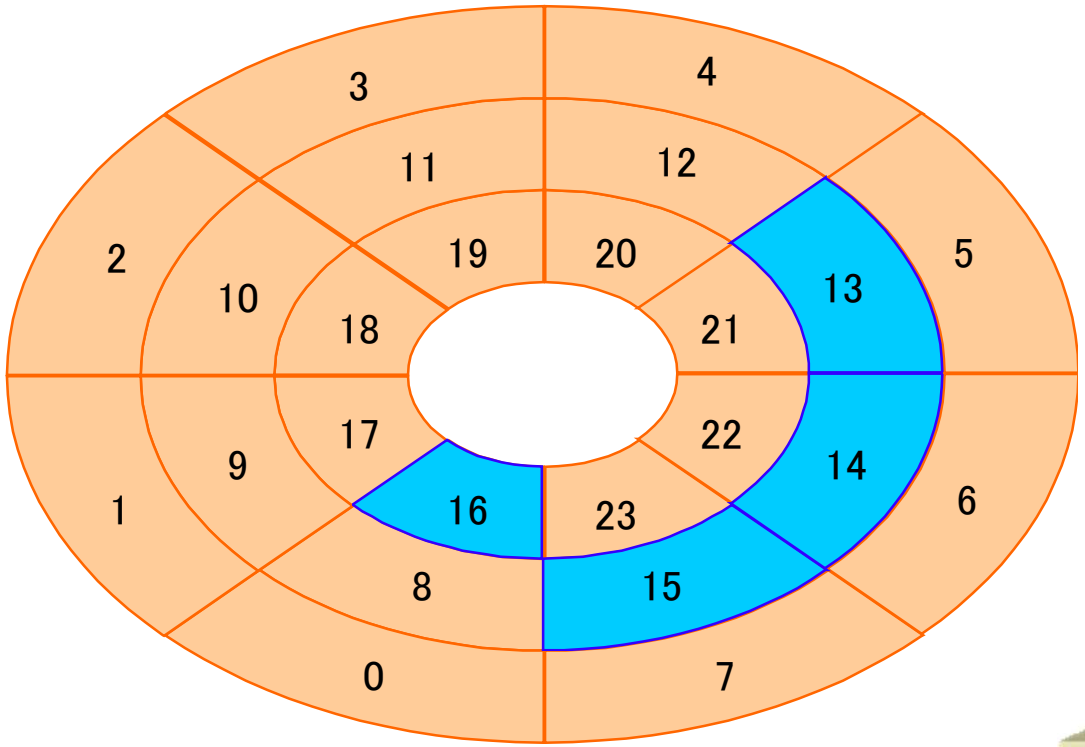
領域割り当て方式

- 固定長割り当て方式
  - → プロセスが必要とする量の二次記憶領域を
    1セクタまたは1クラスタ（複数セクタをまとめた単位）
    単位で割り当てる
  - 複数の単位領域の集合を一領域として扱う仕組が必要
    - → リスト方式
    - → インデクス方式

- 連続領域割り当て方式
  - → プロセス実行前に，
    必要となる大きさの連続したセクタを割り当てる

連続割り当て方式

- 高速なファイルシステムの要求
  - 固定長割り当てでは，複数領域を連結するための再構成オーバヘッドが存在し，アクセス時間が長い
- 連続割り当ての利点
  - ヘッド回転数の減少
  - ファイル内相対位置が求めやすいためランダムアクセスも容易



固定長割当て v.s. 連続領域割当て

固定長割り当て

- ×アクセス時間：長
  - → 複数セクタの再構成
- ○空セクタさえあれば
  いつでも割り当て可
- ○割り当て領域の
  大きさ変更が容易
- ○効率的な割り当て
  （無駄セクタが少）

連続領域割り当て

- ○アクセス時間：短
  - → 連続領域
  - → ランダムアクセスも高速
- ×割り当てはプロセス
  開始時のみ
- ×大きさ変更が困難
- ×領域が余っているのに
  連続していないので、割り
  当てることができない
  ＝＞注意

二種類のディスクフラグメンテーション

- 固定長割り当て時
  - 主にハードディスクにおいて、とびとびのセクターをアクセスするために、ヘッドのシーク時間およびトラック内の当該セクターへの移動時間によるアクセス速度の低下
  - (Windows) デフラグ＝＞連続領域に再割り当て
    - → SSDでは無意味 というより有害
- 連続領域割り当て時
  - 全体的にはハードディスクの容量は十分余っているのに、連続したセクターが割り当てられないことにより、使えない細かな領域のみが残ってしまう
    - → 主記憶領域割り当て時の可変長割り当てと同じ仕組み
      - 但し、セクター単位の固定長領域を割り当てていることに注意が必要

ハードディスクの場合も、セクター単位でしか割り当てられないため、セクター内での余り領域は存在するが、それは固定長割り当てでも、連続領域割り当てでも同じ（主記憶割り当てとは異なることに注意）

まとめ

- 二次記憶
  - テープ型デバイス
    - → 順次アクセスのみ可能
  - ディスク型デバイス
    - → 直接アクセス可能

- ファイル
  - データが存在するセクタ番号に対して，
    ユーザが利用・管理しやすい「ファイル名」による
    アクセスを可能とする仕組み